

Empowering the All Electric Society 

100 years of passion for
technology and innovation

35th
ANNIVERSARY
フエニックス・コンタクト株式会社

Welcome

Python と REST API による PLCnext Runtime 変数の操作

Rev 1.1 (2024年7月4日)



Python からREST API を用いて変数値を取得

Agenda

- 概要
- PLCnext Engineer プロジェクトの用意
- REST API 用 Python モジュールの用意
- Python プログラム：PLCnext Runtime 変数値の読み取り
- Python プログラム：PLCnext Runtime 変数への値書き込み
- 補足：パスワード認証無しでアクセスする場合
- 補足：複数の PLCnext Runtime 変数の操作
- 補足：curl コマンドによる REST API の動作確認

当資料の概要

PLCnext Control※ 上で動いている ST 言語で書かれたプログラム内の変数値を、Python のプログラムから操作する手順をご紹介します。

※ 当資料では AXC F 2152 を例にとっていますが、AXC F 1152/3152, EPC 1502/1522 でも同様の手順となります。

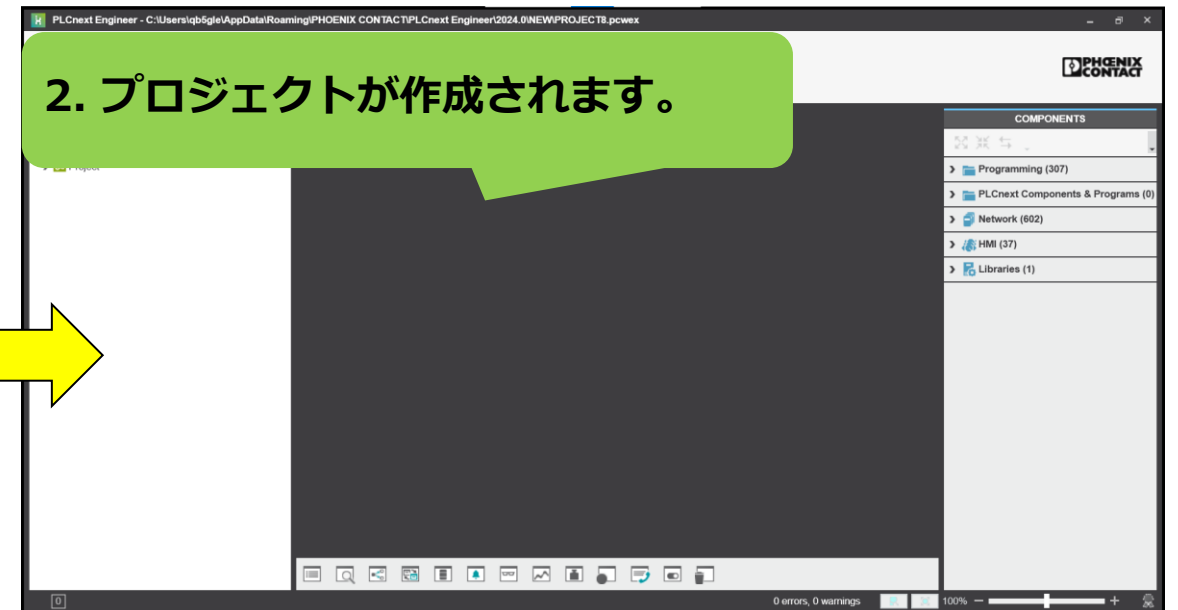
- 簡単なプロジェクトを作成し、プログラム内の変数値を外部から操作するための準備をします。
 - > REST API の機能を使えるように設定を変更します。
 - > 変数 counter_per_task を REST API 経由で外部から操作できるように設定します。
- Python のプログラムから REST API を用いて変数 counter_per_task の値を読み込みます。
- Python のプログラムから REST API を用いて変数 counter_per_task へ値を書き込みます。

- 補足：パスワード認証無しでアクセスする方法を紹介します。
- 補足：複数の変数を操作したいときの Python プログラムの書き方を紹介します。
- 補足：Python を使わず Curl コマンドからも REST API 経由で変数 counter_per_task の値を読めることを確認します。

PLCnext Engineer プロジェクトの用意

PLCnext 用プロジェクトの作成

- パソコン上で PLCnext Engineer を起動し、新しいプロジェクト※を作成してください。
※当資料では、プロジェクト作成の細かい手順や PC と PLCnext の接続設定の詳細については解説しておりません。
[PLCnext Engineer の基本的な操作方法 | PLCnext Technology \(https://plcnnext.jp/archives/53\)](https://plcnnext.jp/archives/53) などの記事をご参照ください。)



※ご利用の PLCnext Control が表示されない場合は (EPC/RFC シリーズ等)、
[EPC1502/1522用プロジェクトの作成 | PLCnext Technology \(https://plcnnext.jp/archives/5069\)](https://plcnnext.jp/archives/5069) をご参照ください。

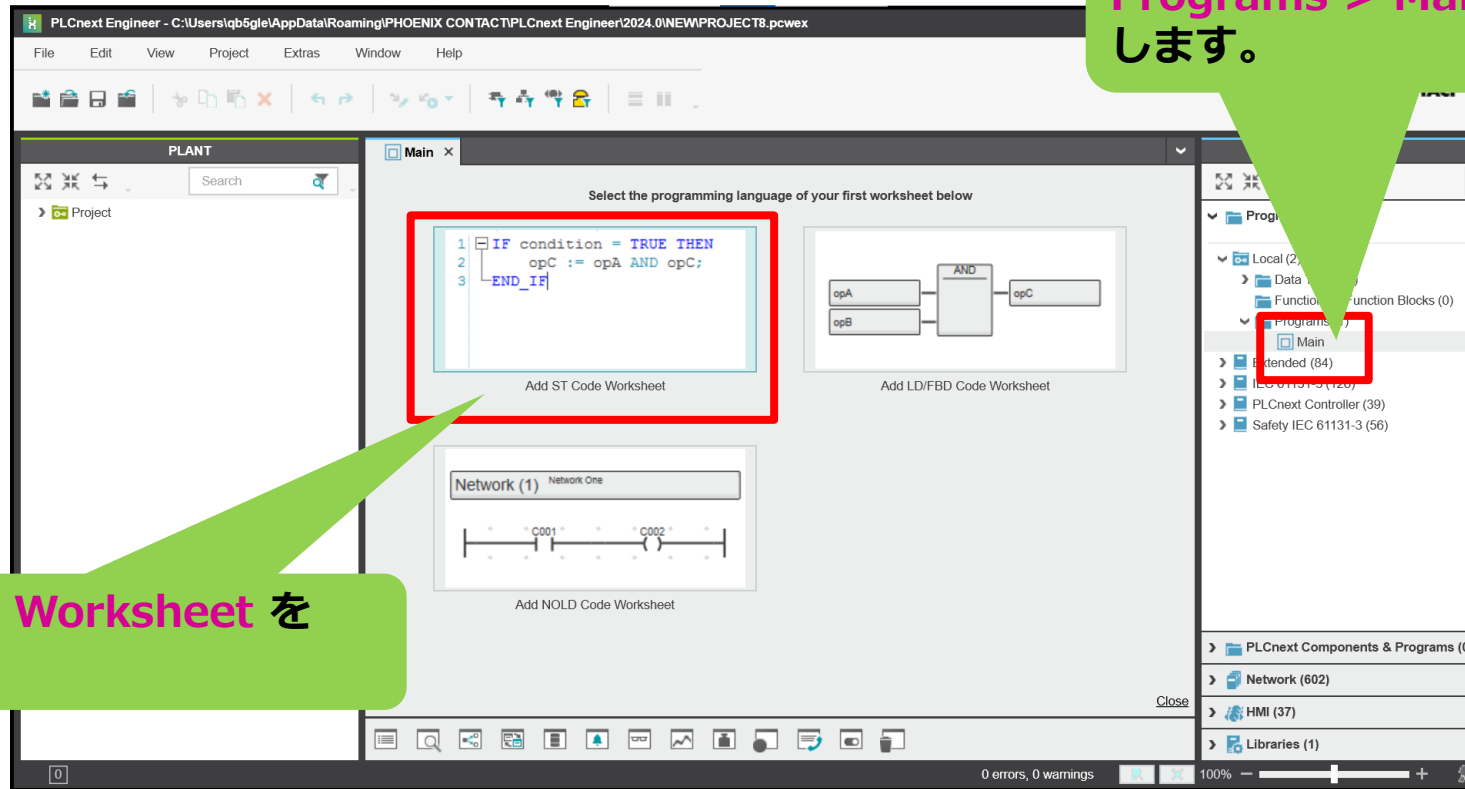
※ご利用の FW バージョンが表示されない場合は、適当な PLCnext Control を選択してプロジェクトを作成してから
[プロジェクトのコントローラ変更方法 | PLCnext Technology \(https://plcnnext.jp/archives/3425\)](https://plcnnext.jp/archives/3425) を参照して Control タイプと FW バージョンがご利用のものと同じように変更してください。

PLCnext Engineer プロジェクトの用意

PLCnext 用プログラムの作成 (1)

- ST Code 用のワークシートを用意します。

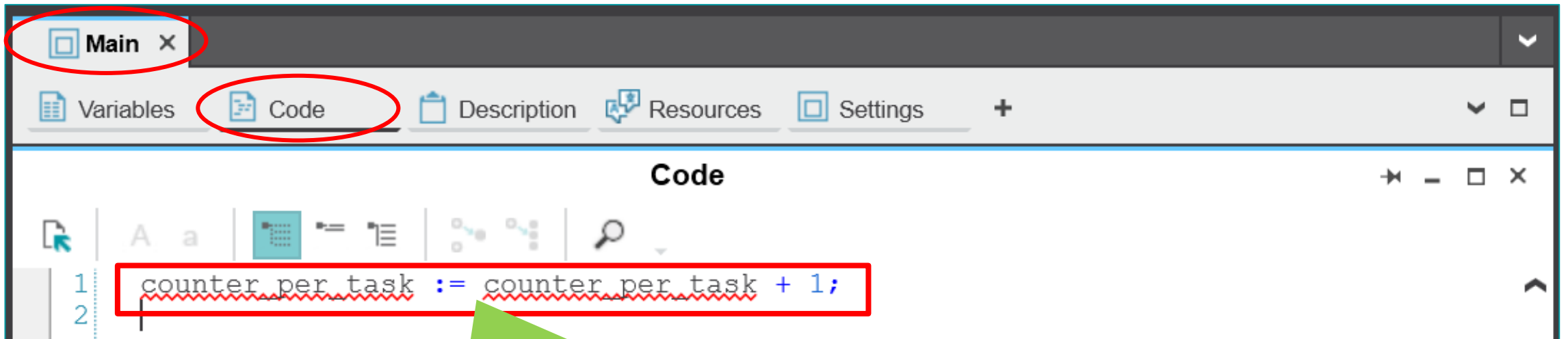
1. Programming > Local > Programs > Main をダブルクリックします。



2. Add ST Code Worksheet をダブルクリック

PLCnext 用プログラムの作成 (2)

- 以下の通り簡単なプログラムを作成してください。



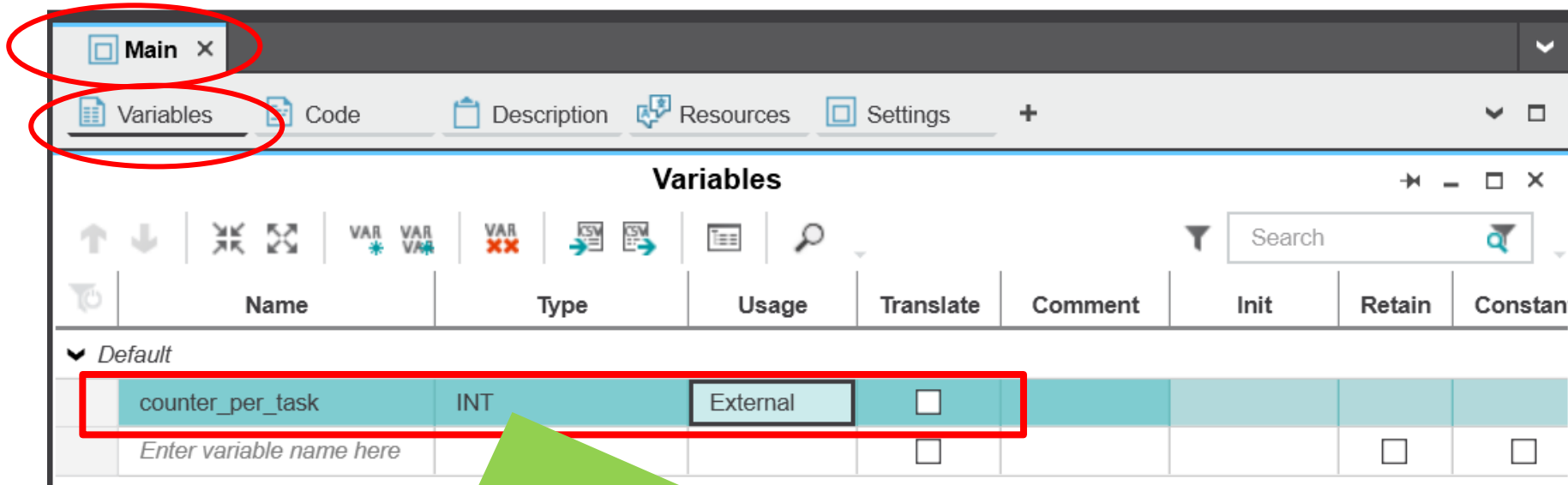
3. Main タブの Code タブを選択し、

```
counter_per_task := counter_per_task + 1;
```

と入力します。

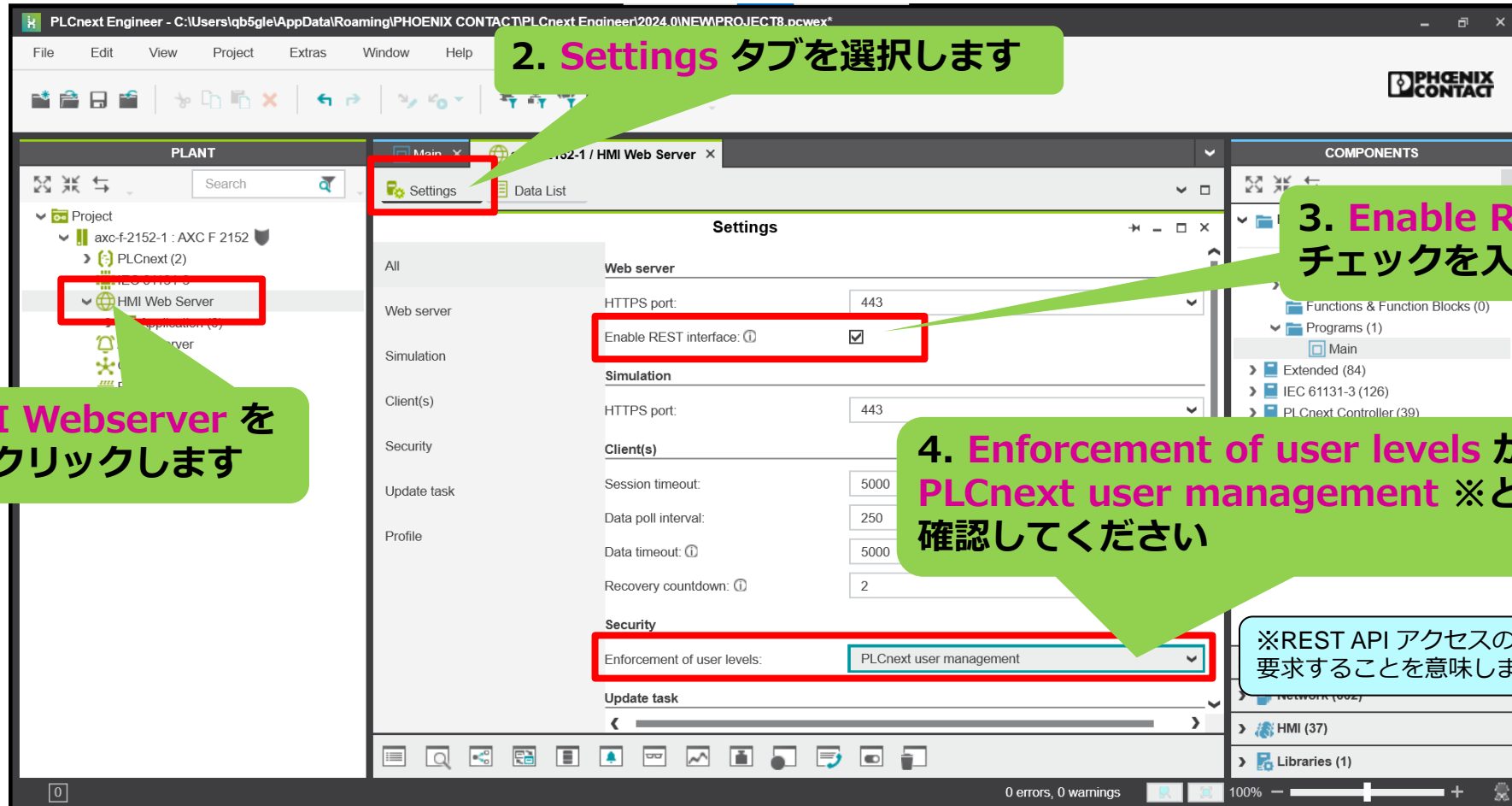
PLCnext 用プログラムの作成 (3)

- Counter_per_task 変数を作成します。



4. Main タブの Variable タブを選択し、counter_per_task という変数を INT 型の External 変数として作成してください。

REST API インターフェースの有効化



変数 counter_per_task の REST API アクセスを許可 (1)

1. axc-f-2152-1 : AXC F 2152 をダブルクリックします

2. Data List タブを選択します。

The screenshot displays the PLCnext Engineer software interface. The left sidebar shows a project tree with 'PLCnext (2)' highlighted. The main workspace is divided into three tabs: 'Cockpit', 'Settings', and 'Data List', with 'Data List' selected. The 'Data List' tab shows a table of variables. The variable 'counter_per_task' is highlighted in the table. The right sidebar shows a 'COMPONENTS' tree with 'Main' selected under 'Programs (1)'. The bottom status bar indicates '0 errors, 0 warnings' and '100%' zoom.

Variable (PLC)	Process data item	HMI tag
PNIO_FORCE_PRIMARY	Profinet / PNIO_FORCE_PRIMARY	
IP_ACTIVE_SOCKETS	Select Process data item here	
TLS_ACTIVE_SOCKETS	Select Process data item here	
HMI_STATUS2	HMI Web Server / HMI_STATUS2	
HMI_CONTROL	HMI Web Server / HMI_CONTROL	
EIPD_INPUTS	Ethernet/IP / EIPD_INPUTS	
EIPD_OUTPUTS	Ethernet/IP / EIPD_OUTPUTS	
EIPD_VALID_DATA_CYCLE	Ethernet/IP / EIPD_VALID_DATA...	
EIPD_PEER_IDLE	Ethernet/IP / EIPD_PEER_IDLE	
EIPD_PEER_RUN	Ethernet/IP / EIPD_PEER_RUN	
EIPD_OUTPUTS_LENGTH	Ethernet/IP / EIPD_OUTPUTS_LE...	
EIPD_INPUTS_LENGTH	Ethernet/IP / EIPD_INPUTS_LEN...	
PLC_CRC_PRJ	Select Process data item here	
counter_per_task	Select Process data item here	
Enter variable name here		

変数 counter_per_task の REST API アクセスを許可 (2)

The screenshot shows the PLCnext Engineer interface. On the left, the 'PLANT' tree view is expanded to 'axc-f-2152-1 : AXC F 2152'. A context menu is open over the 'HMI Web Server' component, with 'Add HMI Tag' highlighted. A green callout box points to this menu item with the text: **4. Add HMI Tag をクリックします。**

In the center, a table lists process data items and HMI tags. The 'counter_per_task' variable is highlighted in the 'Process data item' column. A green callout box points to it with the text: **3. counter_per_task を右クリックします。**

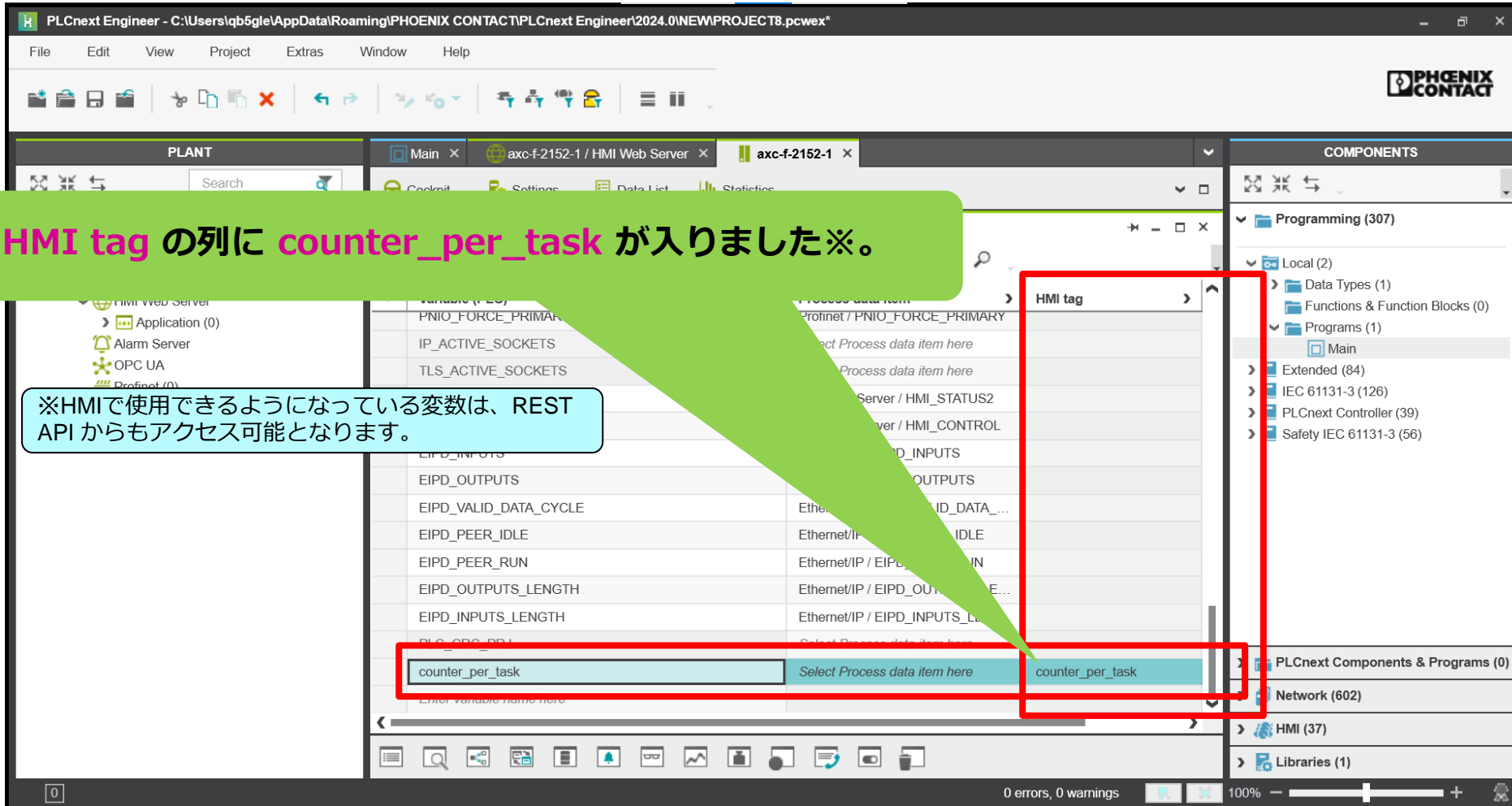
At the bottom, a text input field contains the name 'counter_per_task', which is also highlighted with a red box. Below the input field, it says 'Enter variable name here'.

The right side of the interface shows the 'COMPONENTS' tree view, which includes 'Local (2)', 'Data Types (1)', 'Functions & Function Blocks (0)', 'Programs (1)', 'Main', 'Extended (84)', 'IEC 61131-3 (126)', 'PLCnext Controller (39)', and 'Safety IEC 61131-3 (56)'.

変数 counter_per_task の REST API アクセスを許可 (3)

4. HMI tag の列に counter_per_task が入りました※。

※HMIで使用できるようになっている変数は、REST API からアクセス可能となります。



プロジェクトのビルドと書き込み

1. **axc-f-2152-1 : AXC F 2152** をダブルクリックします

The screenshot shows the PLCnext Engineer interface. On the left, a project tree is visible with 'axc-f-2152-1 : AXC F 2152' selected and highlighted by a red box. A green callout bubble points to this selection with the instruction '1. axc-f-2152-1 : AXC F 2152 をダブルクリックします'. In the center, the 'Cockpit' tab is selected and highlighted by a red box, with a green callout bubble pointing to it and the instruction '2. Cockpit タブをクリックします'. On the right, the 'Cockpit' panel is displayed, showing a 'TCP/IP' dropdown and a download icon (a blue square with a white downward arrow) highlighted by a red box. A green callout bubble points to this icon with the instruction '3. アイコンをクリックします'. Below the download icon, there is a table of diagnostics and status indicators.

Diagnostics and status indicators	
Device	BF-C: ⓘ ●
	BF-D: ⓘ ●
Network	SF: ⓘ ●
PLC runtime	RUN: ●
Notifications	FAIL: ●
	DBG: ●
Ad-hoc IP address	D: ⓘ ●

pyPLCn モジュールのインストール

- Python から PLCnext Control の REST API を簡単に扱えるように、モジュール「pyPLCn」を pip※でインストールします。

```
$ su  
$ pip install pyPLCn  
$ exit
```

※ pip を使用するには PLCnext Control をインターネットへ接続する必要があります。
下記のサイト等をご参照ください。

[AXC F 1152/2152/3152をインターネットへ接続する | PLCnext Technology \(https://plcnext.jp/archives/4051\)](https://plcnext.jp/archives/4051)

※ su コマンドを使用する際、root ユーザにパスワードを設定しておく必要があります。(デフォルトでは設定されていません)
下記のサイト等をご参照ください。

[Linux root ユーザー作成 | PLCnext Technology \(https://plcnext.jp/archives/1610\)](https://plcnext.jp/archives/1610)

※ pip はデフォルトではインストールされていません。
以下のURLの手順に従ってインストールしてください。

<https://plcnext.jp/wp-content/uploads/PIP-インストール手順curl.pdf>

変数値を読み出す Python コードの作成

- 以下のコードを作成し、`read_counter.py` という名前で PLCnext コントローラへ保存してください。

```
from pyPLCn import pyPLCn
from time import sleep

if __name__ == '__main__':
    Plc = pyPLCn()
    Plc.set_var_names(['counter_per_task'])
    Plc.connect( 'localhost', login = 'admin', password = '<実際のパスワード>', poll_time = 100)

    while True:
        print( 'counter_per_task = {}'.format(Plc.get_var('counter_per_task')) )
        sleep(1)
```

① REST API の接続先の PLCnext Runtime が動作している PLCnext Control の IP アドレスです。

② PLCnext Control にデフォルトで存在するユーザアカウント “admin” を指定しています。

③ admin ユーザ用の**実際のパスワード**を入力してください。デフォルトパスワードは PLCnext Control の筐体表面に印字されています。

Python プログラム : PLCnext Runtime 変数値の読み取り 変数値の読み出しを実行

- `read_counter.py` を実行します。
\$ `python3 ./read_counter.py`

```
admin@axcf2152:~$ python3 read_counter.py
INFO:root:Session ID = s2284032
INFO:root:Client State = 3d0ed4ac9ce5ad27
INFO:root:Auth code = 2b4f18ade0aaa3b1
INFO:root:Access token = c5c353a24d3f38ba728de6b73594d639
INFO:root:Vars group = g1284193
counter_per_task = None
counter_per_task = 2377
counter_per_task = 2386
counter_per_task = 2397
```

`counter_per_task` 変数の値が表示され、PLCnext Runtime の変数値を読み出せていることがわかります。

- 中断するときは、Ctrl + x を押してください。

変数値を書き込む Python コードの作成

- 以下のコードを作成し、`set_counter.py` という名前で PLCnext コントローラへ保存してください。

```
from pyPLCn import pyPLCn
from time import sleep

if __name__ == '__main__':
    Plc = pyPLCn()
    Plc.set_var_names(['counter_per_task'])
    Plc.connect( 'localhost', login = 'admin', password = '<実際のパスワード>', poll_time = 100)

    debug_counter = 0
    while True:
        if debug_counter > 10:
            Plc.set_var('counter_per_task', '0')
            print ('reset')
            debug_counter = 0
        print( 'counter_per_task = {}'.format(Plc.get_var('counter_per_task')) )
        sleep(1)
        debug_counter = debug_counter + 1
```

実際のパスワードを入力してください

紫の部分は、`read_counter.py` からの差分 (追加のみ) です。

Python プログラム : PLCnext Runtime 変数への値書き込み 変数値の書き込みを実行

- `set_counter.py` を実行します。

```
$ python3 ./set_counter.py
```

```
counter_per_task = 90  
counter_per_task = 100  
reset  
counter_per_task = 110  
counter_per_task = 9  
counter_per_task = 19  
counter_per_task = 30  
counter_per_task = 39  
counter_per_task = 50  
counter_per_task = 59  
counter_per_task = 70  
counter_per_task = 79  
counter_per_task = 90  
counter_per_task = 99  
reset  
counter_per_task = 110  
counter_per_task = 9  
counter_per_task = 19
```

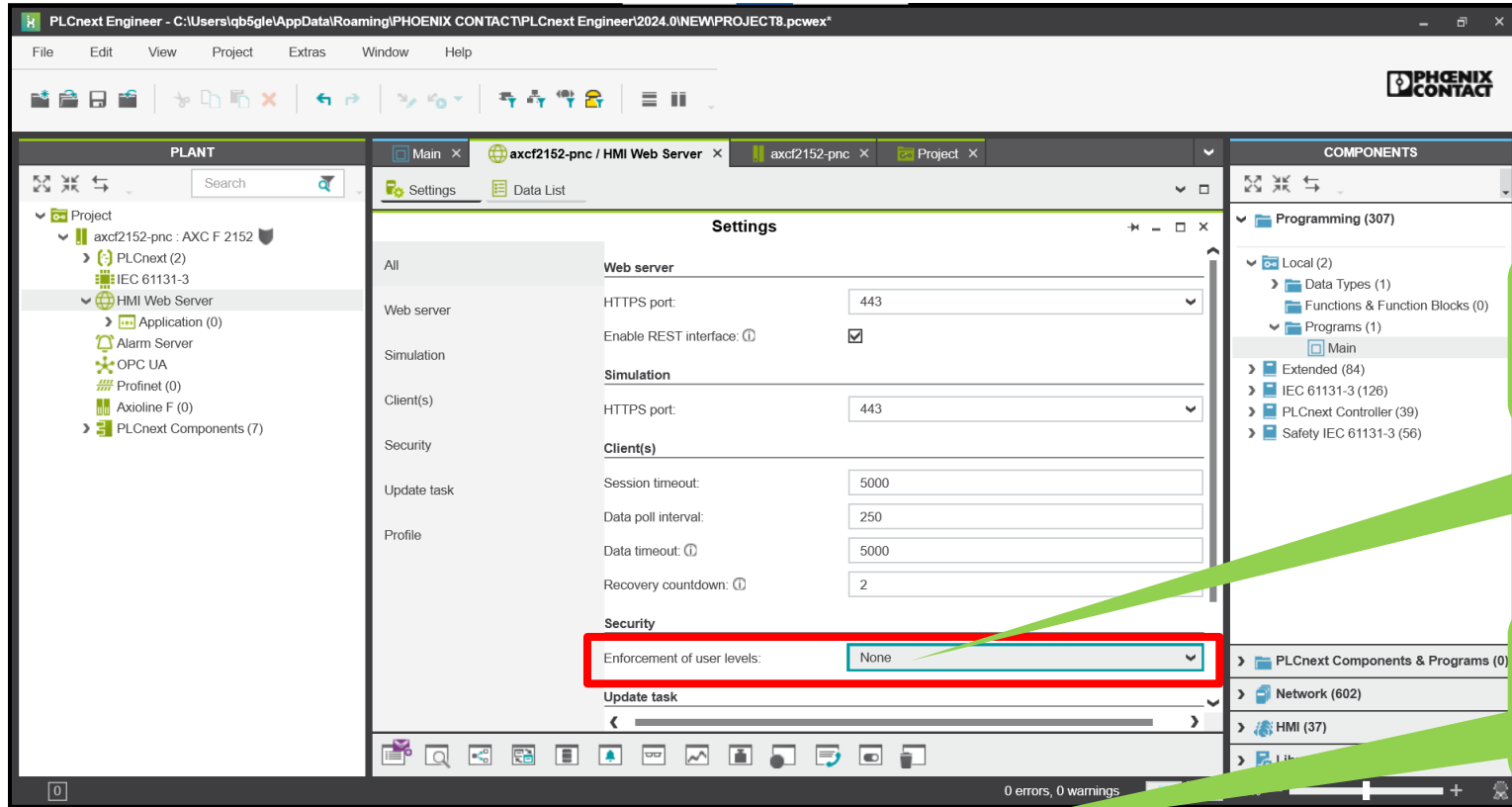
`reset`※ が表示された後に、`counter_per_task` の値が小さくなっています。
つまり、REST API による PLCnext Runtime 変数値の更新が行われたことがわかります。

※ REST API によるアクセス結果が反映されるまでには、若干のディレイがあります。

- 中断するときは、Ctrl + x を押してください。

補足

パスワード認証無しでアクセスする場合



1. Enforcement of user levels が None であれば、認証は不要です。

2. Python コード側は、login と password へ空文字列を指定します。

```
Plc.connect( 'localhost', login = "", password = "", poll_time = 100)
```

補足

複数の PLCnext Runtime 変数の操作

- REST API でアクセスする変数は、set_var_names 関数の引数の配列内にあらかじめ変数名を全て登録しておく必要があります。

```
Plc.set_var_names(['counter_per_task', 'second_val'])
```

- PLCnext Engineer のプロジェクト側にも該当する変数を用意し、REST API でアクセスできるように HMI_tag を付与してください。

補足

curl コマンドによる REST API の動作確認

- REST API でアクセスできる変数の一覧を取得

```
$ curl --insecure https://192.168.1.10/ehmi/data.dictionary.json
```

```
192.168.1.10 - root@48539c77c436: / VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
admin@axcf2152:~$ curl --insecure https://192.168.1.10/ehmi/data.dictionary.json
{"$schema":"http://json-schema.org/draft-04/schema#","title":"DataDictionary","HmiVariables2":{"Arp.Plc.Eclr/ESM_DATA.ESM_INFOS[1].TICK_COUNT":{"Type":"DINT","InitValue":"0","ReadOnly":true},"Arp.Plc.Eclr/counter_per_task":{"Type":"INT","InitValue":"INT#0"}}}admin@axcf2152:~$
```

counter_per_task 変数には Arp.Plc.Eclr/counter_per_task という名前でアクセスできることがわかります。

- counter_per_task 変数の値を取得

```
$ curl --insecure https://192.168.1.10/_pxc_api/api/variables?paths=Arp.Plc.Eclr/counter_per_task
```

```
admin@axcf2152:~$ curl --insecure https://192.168.1.10/_pxc_api/api/variables?paths=Arp.Plc.Eclr/counter_per_task
{"apiVersion":"1.10.0.0","projectCRC":2613350864,"userAuthenticationRequired":true,"variables":[{"path":"Arp.Plc.Eclr/counter_per_task","value":20623}]}admin@axcf2152:~$
admin@axcf2152:~$ curl --insecure https://192.168.1.10/_pxc_api/api/variables?paths=Arp.Plc.Eclr/counter_per_task
{"apiVersion":"1.10.0.0","projectCRC":2613350864,"userAuthenticationRequired":true,"variables":[{"path":"Arp.Plc.Eclr/counter_per_task","value":20686}]}admin@axcf2152:~$
```

counter_per_task 変数の値を確認することができます。